

FIG.1

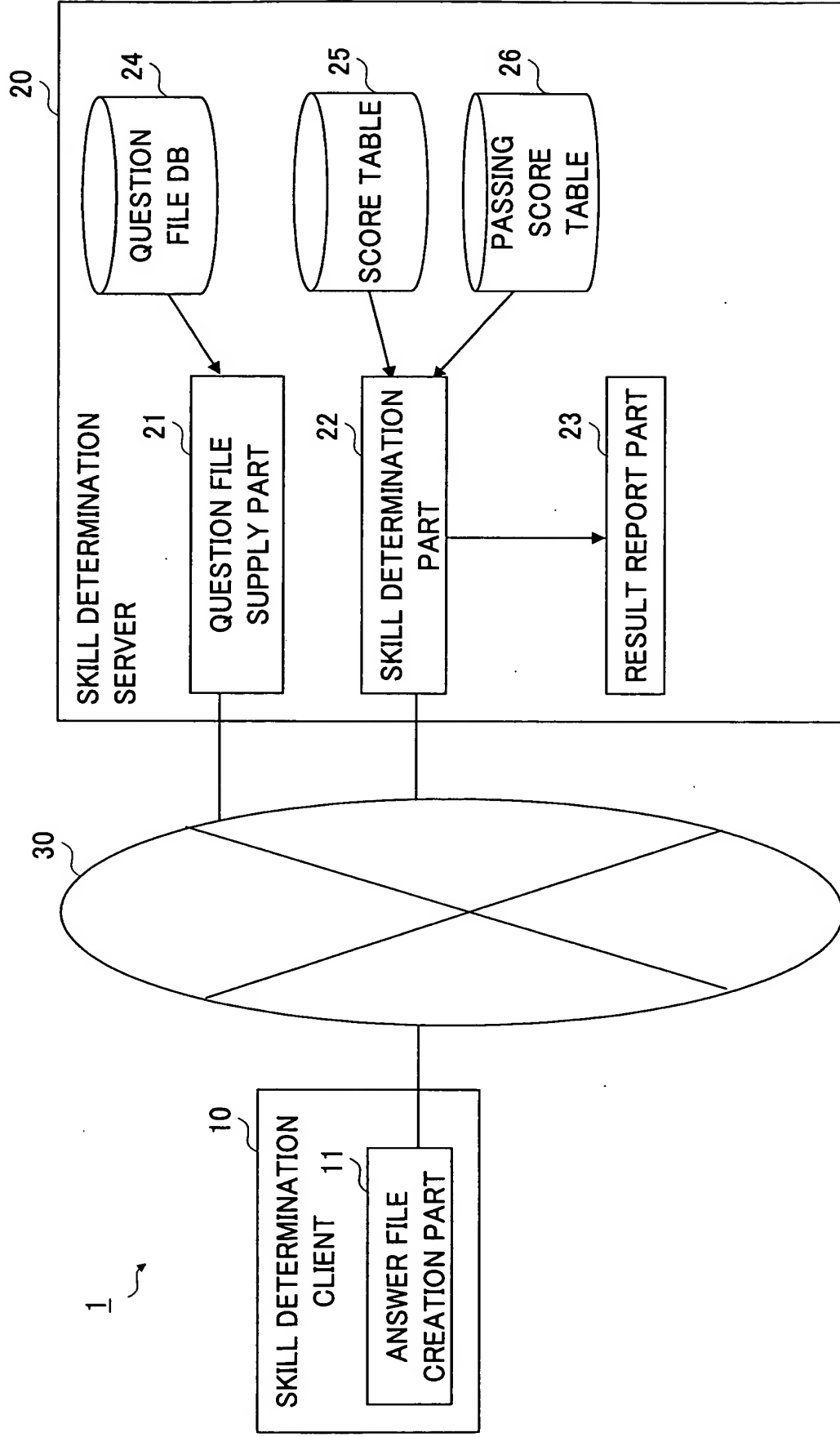


FIG.2

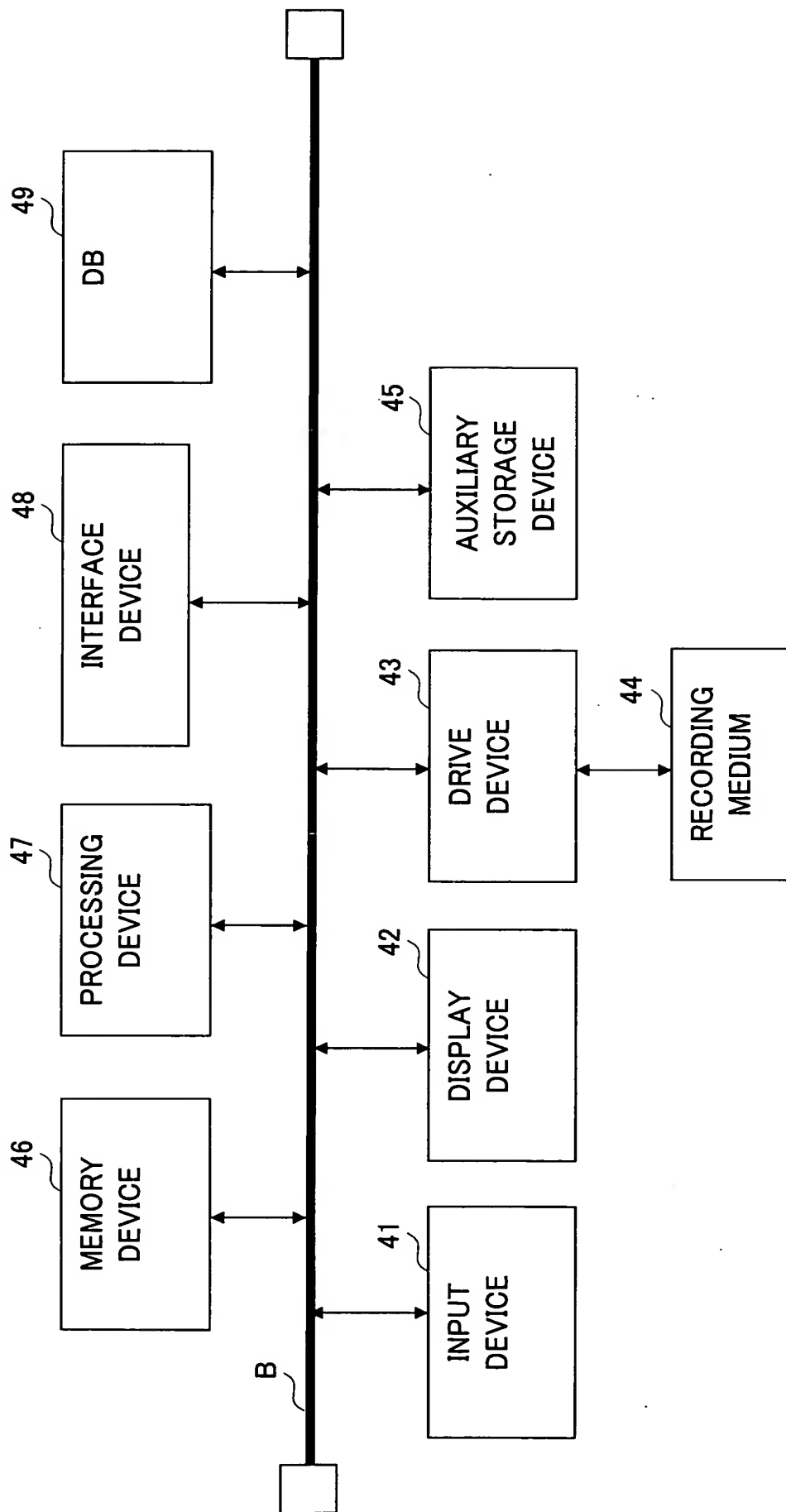


FIG.3

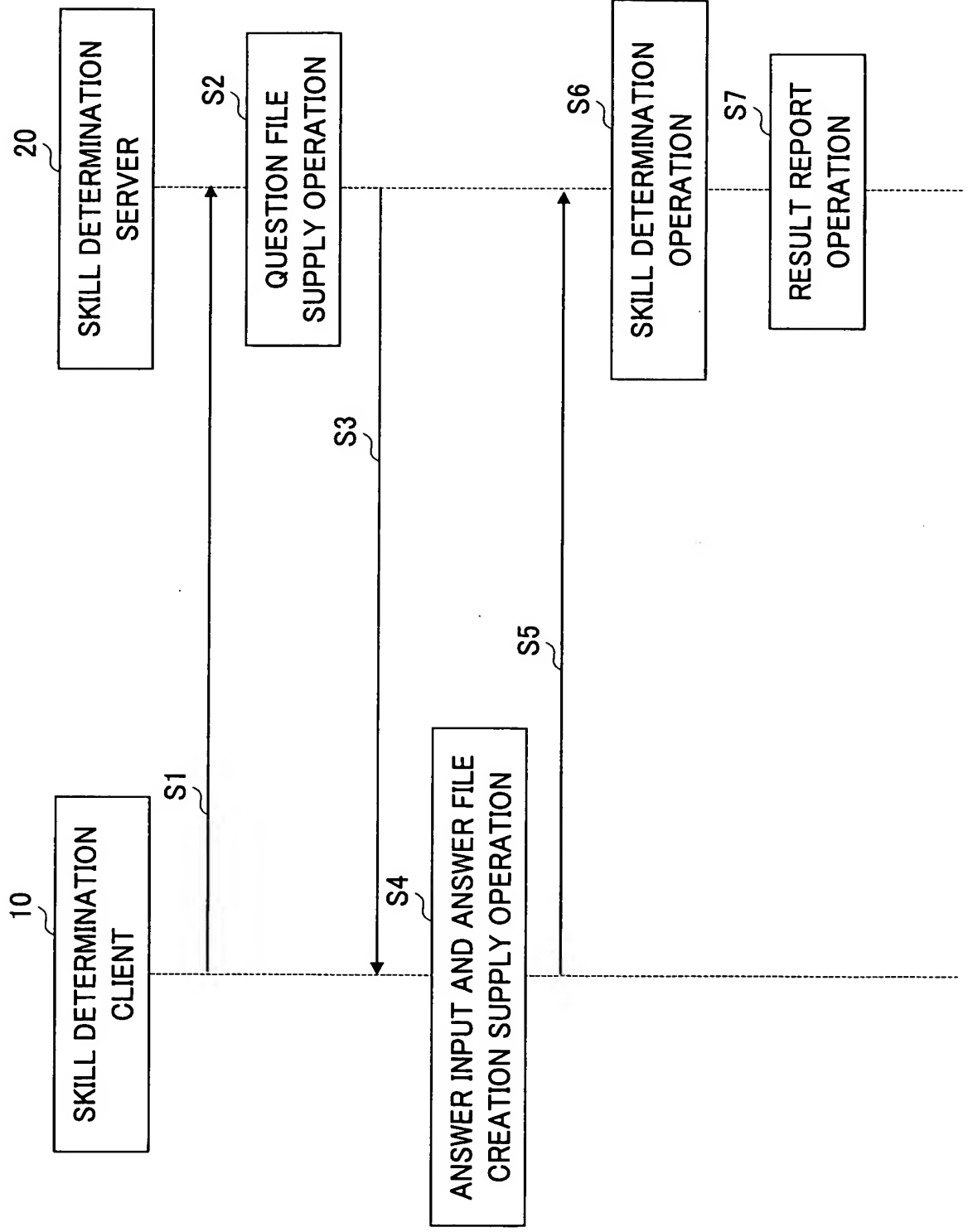


FIG.4

ASSIGNMENT

INSTRUCTION: Design a circuit for lighting ON/OFF LED by manipulating a push button after an input signal CE is asserted by filling in empty spaces in the following circuit source code (Verilog-HDL). (Note that LED is OFF before the input signal CE is asserted.)

```

module light_led(
    clk,                // 33MHz clock input signal
    rst_n,              // reset input signal (negative logic)
    ce,                 // chip enable input signal
    push_n,             // push button input signal (negative logic)
    led_n               // light ON LED (negative logic) output signal
);

    // Declaration part of input and output port signals
    input      clk;
    input      rst_n;
    input      ce;
    input      push_n;

    output     led_n;

    // Declaration part of inner signal lines and registers
    reg [1] r_cnt;      // 20 bit counter register (for generation of chattering prevention clock)
    wire     s_cnt_end; // 20 bit counter end signal (for generation of chattering prevention clock)
    reg      r_pc_lk;   // chattering prevention clock signal register
    reg      r_push_n;  // signal register for keeping the current push button signal with respect to r_pc_lk synchronization
    reg      r_pre_push_n // signal register for keeping the previous push button signal with respect clock synchronization
    wire     s_pushed;  // chattering prevented push button input signal

    reg [2] r_state;    // 3 bit width register indicative of the current circuit status
    wire     s_state_idle; // signal line indicative of an idle status
    wire     s_state_led_off; // signal line indicative of a LED light OFF status
    wire     s_state_led_on; // signal line indicative of a LED light ON status

```

FIG.5

```
// Register statuses as parameters
parameter IDLE = 3; // chip enable input signal wait status
parameter LED_OFF = 4; // LED light OFF status
parameter LED_ON = 5; // LED light ON status

// Combinatorial circuit description
/* If the counter is "0xF_FFFF", output "1", otherwise, output "0".
(Logical operations for all bits are used.) */
assign s_cnt_end = 6 r_cnt;

/* Only if the push button is pushed, assign "1". */
assign s_pushed = ~ 7 & 8;

assign s_state_idle = 9;
assign s_state_led_off = 10;
assign s_state_led_on = 11;

assign led_n = 12;

// Order circuit description
/* Count up in a status other than the idle status. */
always @( 13 ) begin
/* If reset becomes active, input "0" as an initial value. */
if ( rst_n == 1'b0 ) begin
    r_cnt <= 14;
end else begin
```

FIG.6

```

if ( [ 15 ] ) begin
    [ 16 ] ;
end
end
end

/* Generate chattering prevention clock. */
always @( [ 17 ] ) begin
    if ( rst_n == 1'b0 ) begin
        r_pclk <= 1'b0;
    end else begin
        if ( [ 18 ] ) begin
            r_pclk <= [ 19 ] ;
        end
    end
end
end

/* Keep the current push button input in a status other than the idle status. */
always @( [ 20 ] ) begin
    if ( rst_n == 1'b0 ) begin
        r_push_n <= 1'b1;
    end else begin
        if ( [ 21 ] ) begin
            r_push_n <= [ 22 ] ;
        end
    end
end
end

/* Keep the previous push button input in a status other than the idle status. */
always @( [ 23 ] ) begin
    if ( ~rst_n ) begin
        r_pre_push_n <= 1';
    end else begin
        if ( [ 24 ] ) begin
            r_pre_push_n <= [ 25 ] ;
        end
    end
end
end
end

```

FIG.7

```

/* State-machine portion */
always @( [ 26 ] ) begin
    if ( rst_n == 1'b0 ) begin
        r_state <= IDLE;
    end else begin
        [ 27 ]
        IDLE :
            begin
                if ( [ 28 ] ) begin
                    r_state <= LED_OFF;
                end
            end
        LED_OFF : begin
            if ( [ 29 ] ) begin
                r_state <= LED_ON;
            end
        end
        LED_ON : begin
            if ( [ 30 ] ) begin
                r_state <= LED_ON;
            end
        end
        default : begin
            [ 31 ] ;
        end
        [ 32 ]
    end
end
[ 33 ]

```

FIG.8

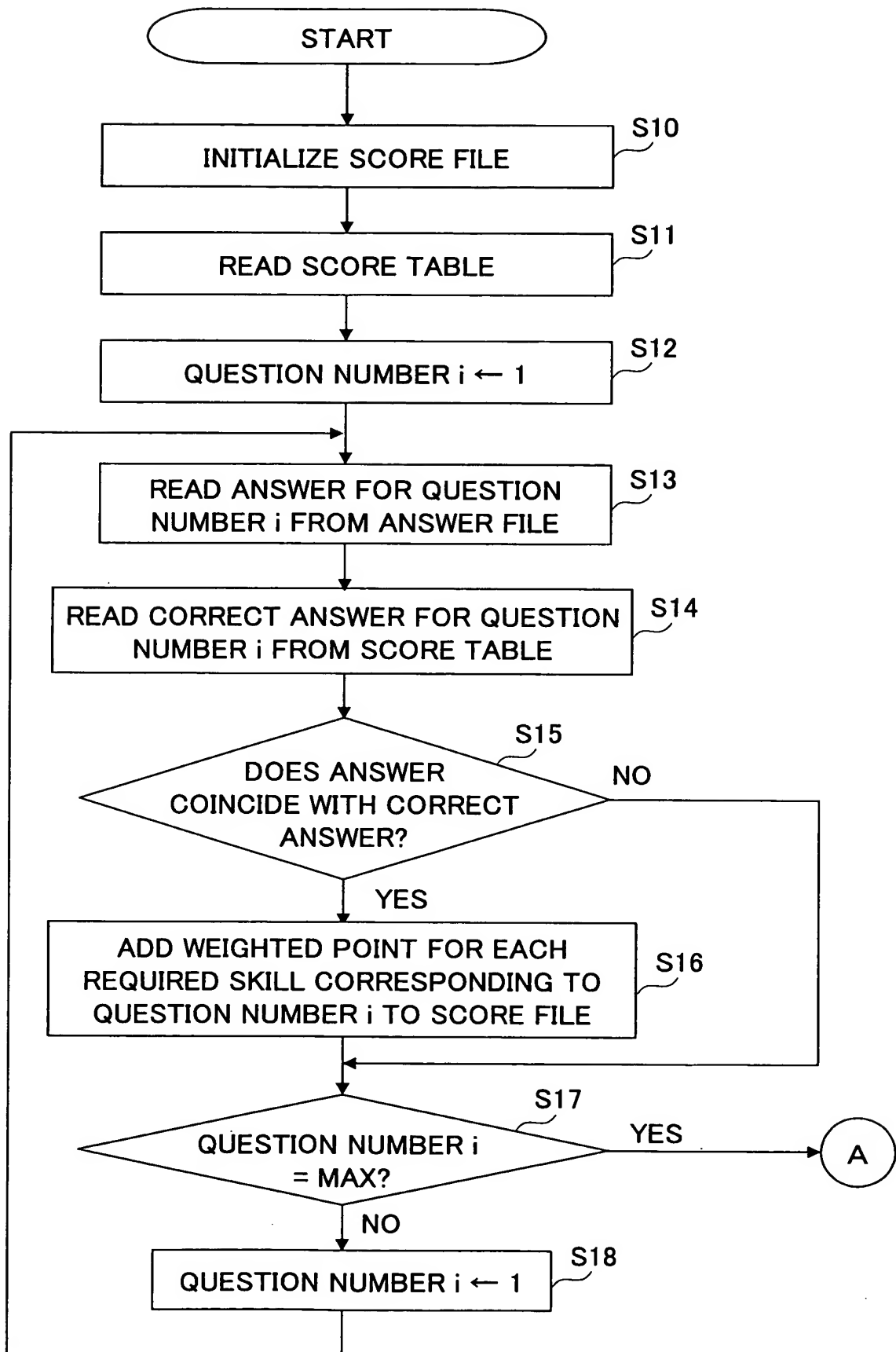


FIG.9

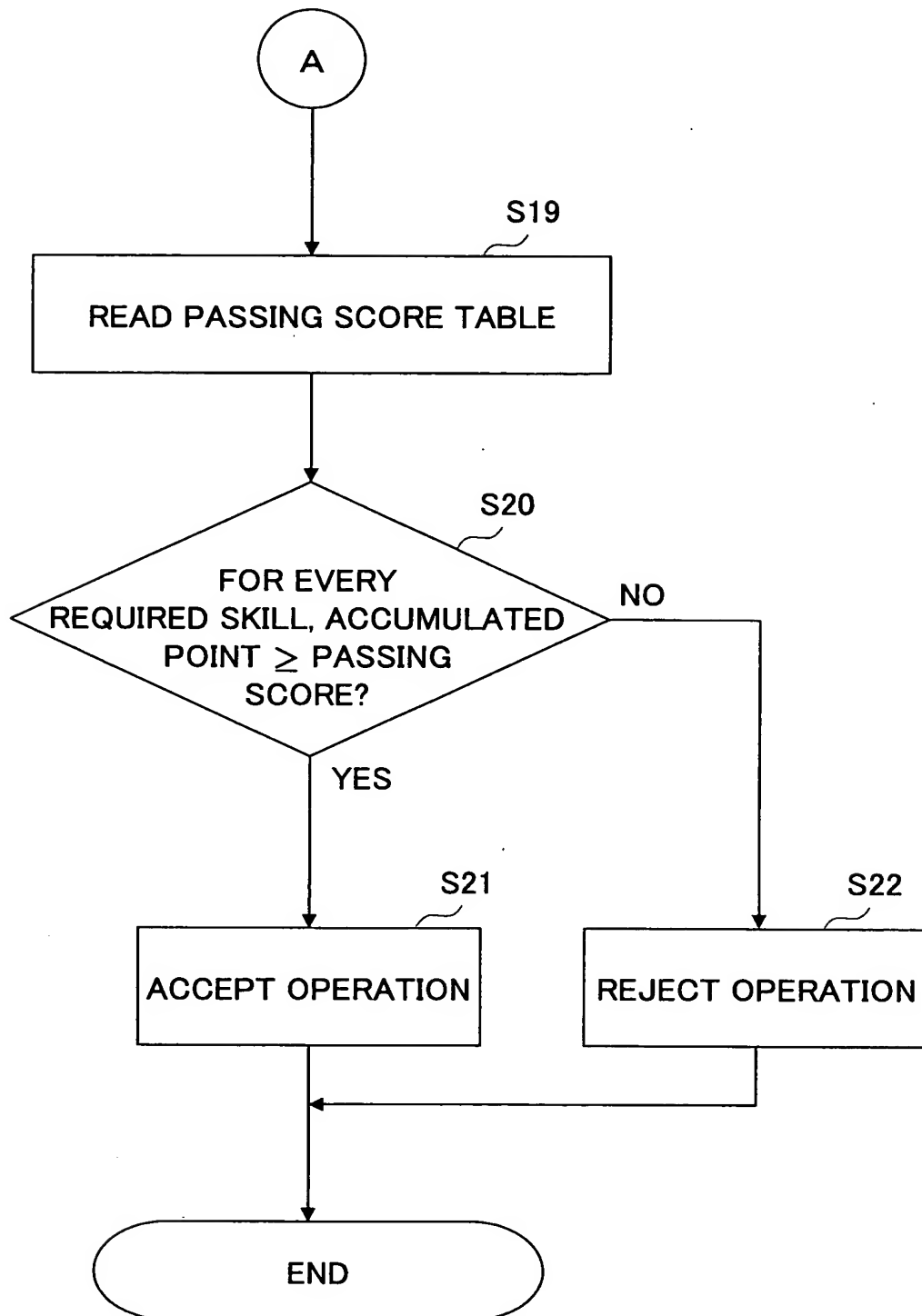


FIG.10

QUESTION NUMBER	CORRECT ANSWER	REQUIRED SKILLS (WEIGHTED POINTS)						SEQUENTIAL CIRCUIT DESCRIPTION
		SPECIFICATION COMPREHENSION	HDL GRAMMAR	SIGNAL DEFINITION	PARAMETER DEFINITION	COMBINATIONAL CIRCUIT DESCRIPTION		
1	19:0	1	1	1				
2	2:0	1	1	1				
3	3'b001	1	1		2			
4	3'b010	1	1		2			
5	3'b100	1	1		2			
6	&	2	1			2		
7	r_push_n	2	1			1		
8	r_pre_push_n	2	1			1		
9	r_state[0]	1	1		1	1		
10	r_state[1]	1	1		1	1		
11	r_state[2]	1	1		1	1		
12	s_state_led_on	3	1			2		
13	posedge clk or negedge rst_n	1	1				1	
14	20'h0	1	1				1	
15	s_state_idle	2	1				2	
16	r_cnt <= r_cnt + 20'h1	3	1				2	
17	posedge clk or negedge rst_n	1	1				1	
18	s_cnt_end	2	1				2	
19	r_pclk	3	1				2	
20	posedge r_pclk or negedge rst_n	1	1				1	
21	s_state_idle	2	1				2	
22	push_n	3	1				2	
23	posedge clk or negedge rst_n	1	1				1	
24	s_state_idle	2	1				2	
25	r_push_n	3	1				2	
26	posedge clk or negedge rst_n	1	1				1	
27	case (r_state)	1	1				2	
28	ce	2	1				2	
29	s_pushed	2	1				2	
30	s_pushed	2	1				2	
31	r_state <= IDLE	3	1				2	
32	endcase	1	1				1	
33	endmodule	1	1				1	

FIG.11A

61

REQUIRED SKILLS						
	SPECIFICATION COMPREHENSION	HDL GRAMMAR	SIGNAL DEFINITION	PARAMETER DEFINITION	COMBINATIONAL CIRCUIT DESCRIPTION	SEQUENTIAL CIRCUIT DESCRIPTION
MAXIMUM SCORE	55	33	2	9	6	34
PASSING SCORE	45	28	1	6	6	27

FIG.11B

62

REQUIRED SKILLS						
	SPECIFICATION COMPREHENSION	HDL GRAMMAR	SIGNAL DEFINITION	PARAMETER DEFINITION	COMBINATIONAL CIRCUIT DESCRIPTION	SEQUENTIAL CIRCUIT DESCRIPTION
SCORE	42	27	2	9	5	29

FIG.12

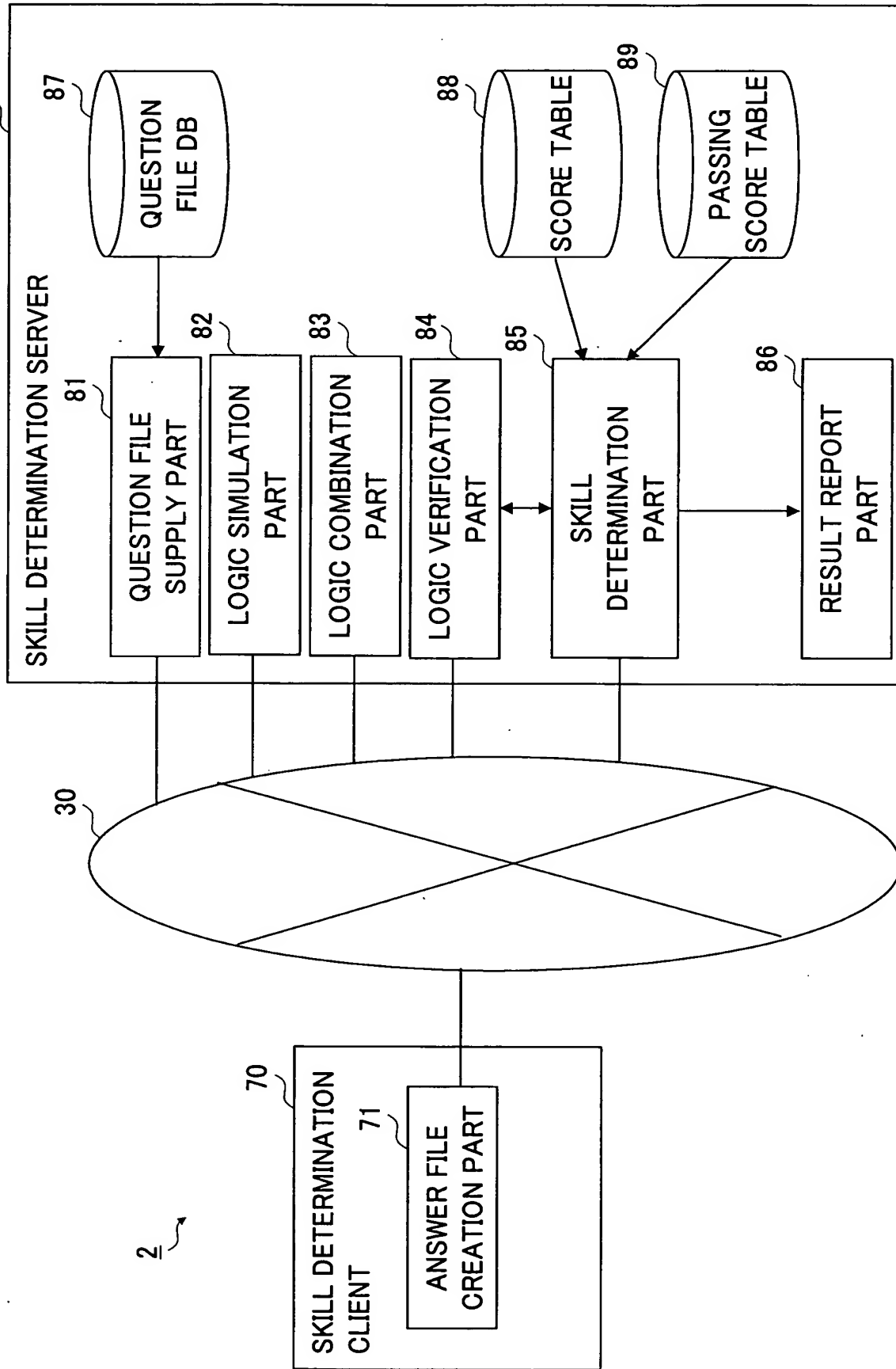


FIG.13

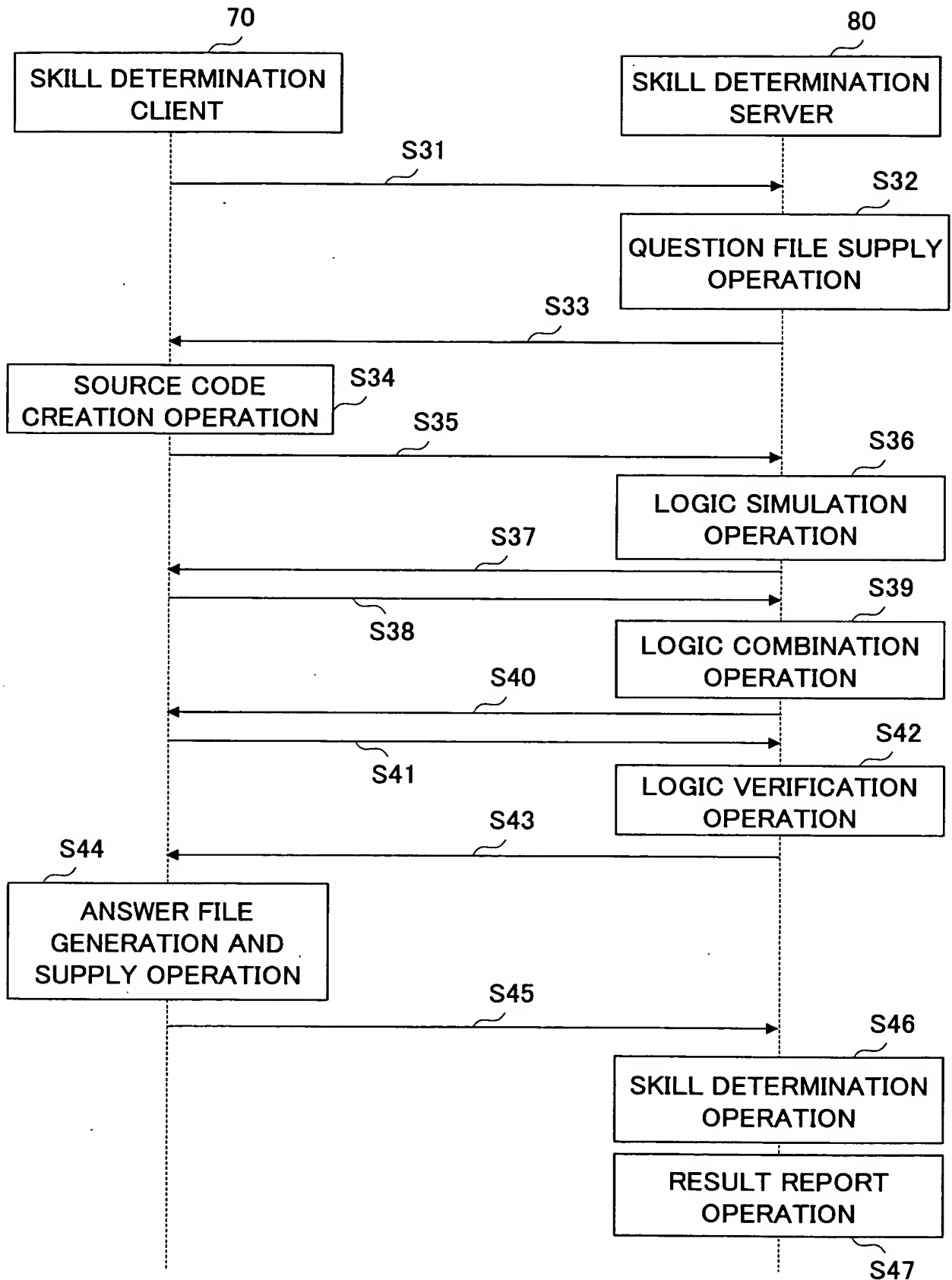


FIG.14

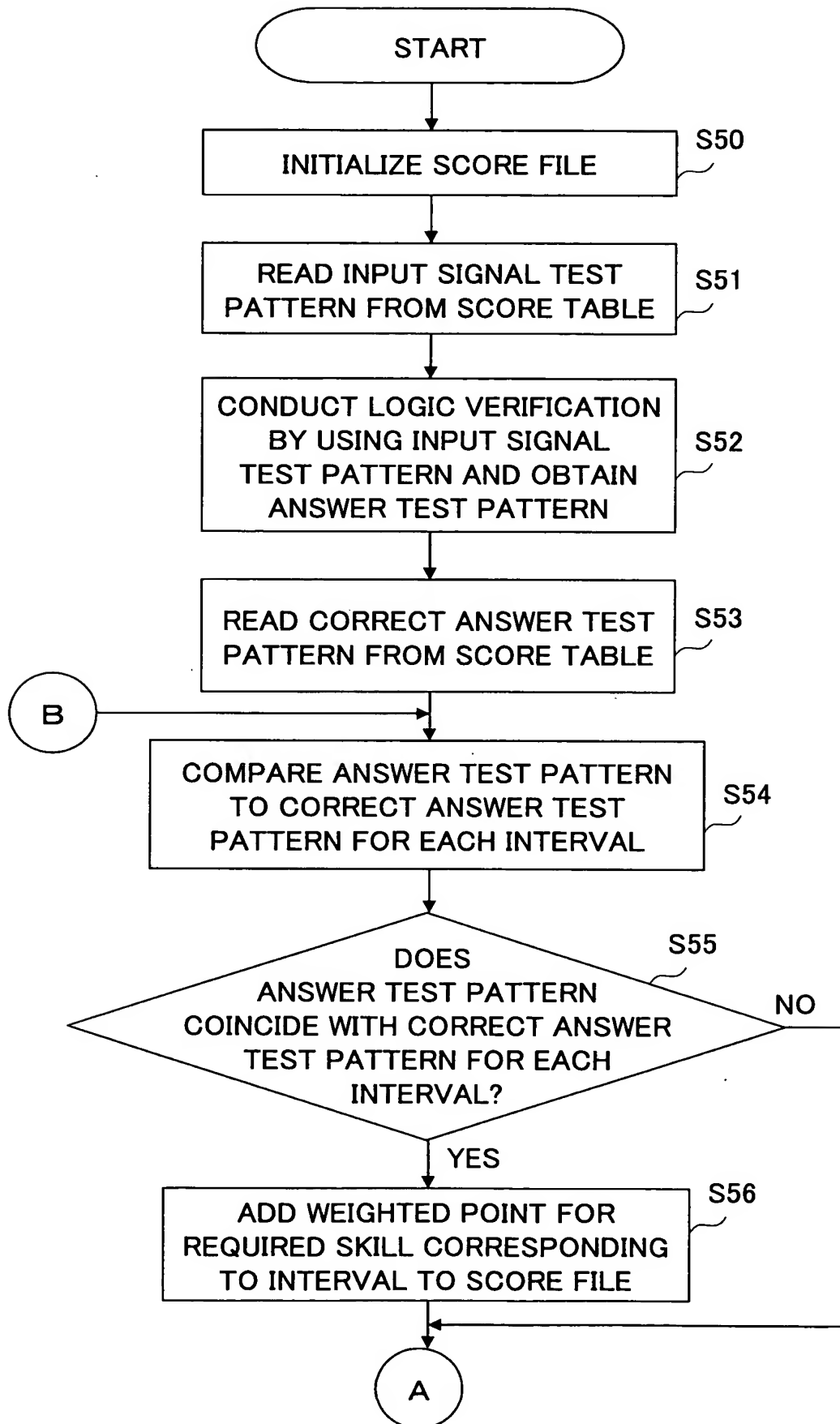


FIG.15

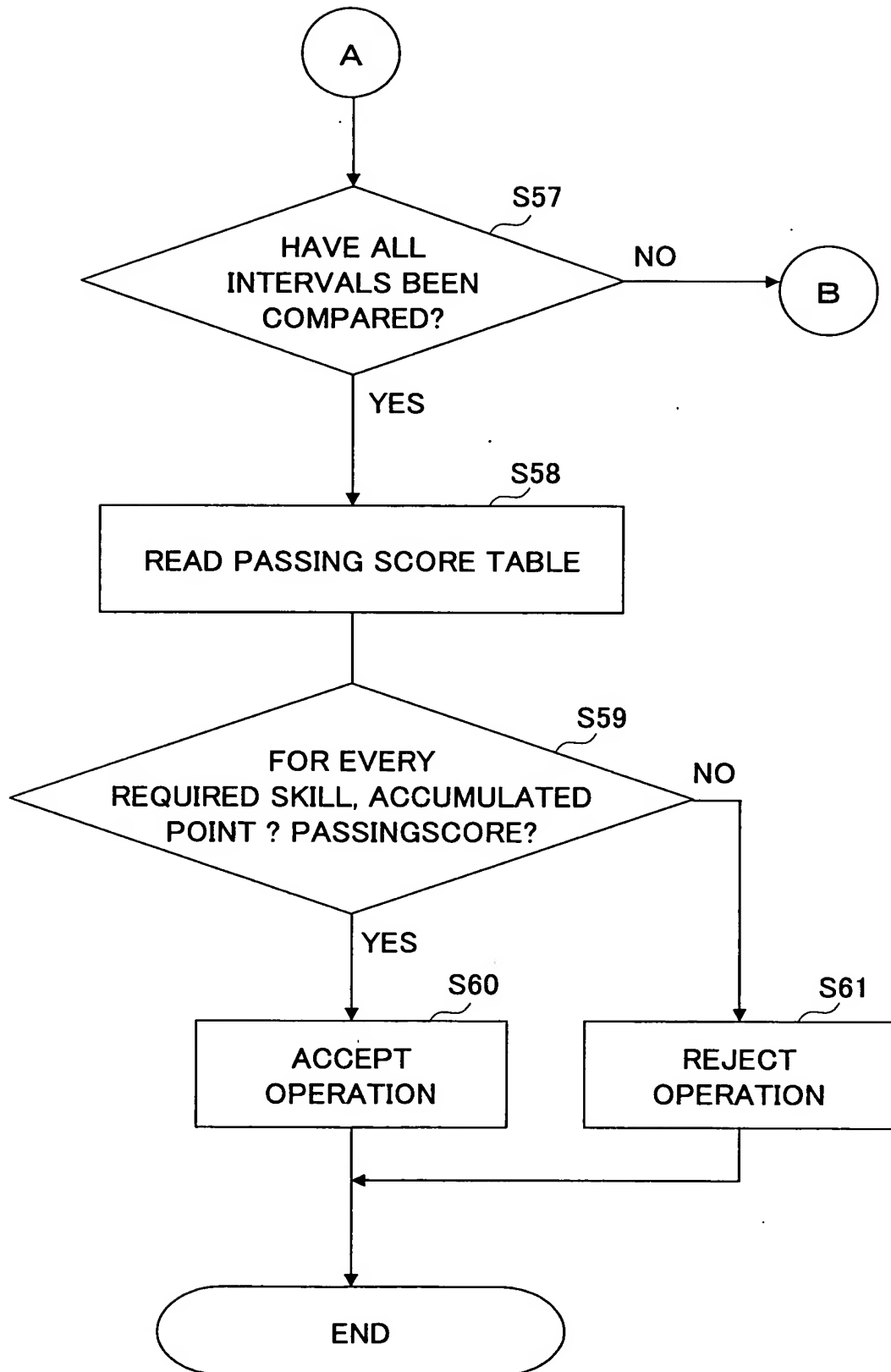


FIG.16

	SIGNAL NAMES
INPUT SIGNAL	CLK
	RST_N
	CE
	PUSH_N
CORRECT ANSWER OUTPUT SIGNAL	LED_N
	INTERVAL
ANSWER OUTPUT SIGNAL	LED_N

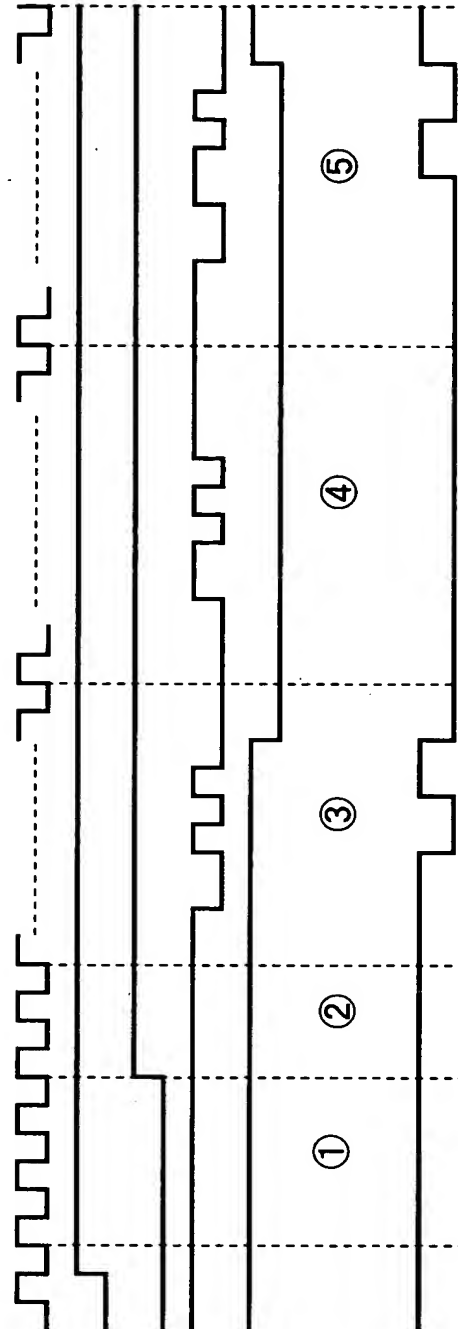


FIG.17

INTERVALS	NOR OPERATION RESULTS	COINCIDENCE DEGREES BETWEEN WAVEFORM START AND END POINTS	NUMBER OF WAVEFORM CHANGES
①	3	—	—
②	2	—	—
③	—	1	3
④	—	1	0
⑤	—	1	3

FIG.18

INTERVALS	REQUIRED SKILLS				
	SPECIFICATION COMPREHENSION	RESET OPERATION	PUSH BUTTON OPERATION	LED OPERATION	CHATTERING COMPREHENSION
①、②	1	1		1	
③、④、⑤	*6		*7	*7	*6

FIG.19A

91

REQUIRED SKILLS					
	SPECIFICATION COMPREHENSION	RESET OPERATION	PUSH BUTTON OPERATION	LED OPERATION	CHATTERING COMPREHENSION
MAXIMUM SCORE	5	1	1	2	4
PASSING SCORE	3	1	1	2	2

FIG.19B

92

REQUIRED SKILLS					
	SPECIFICATION COMPREHENSION	RESET OPERATION	PUSH BUTTON OPERATION	LED OPERATION	CHATTERING COMPREHENSION
SCORE	3	1	1	2	2

FIG.20

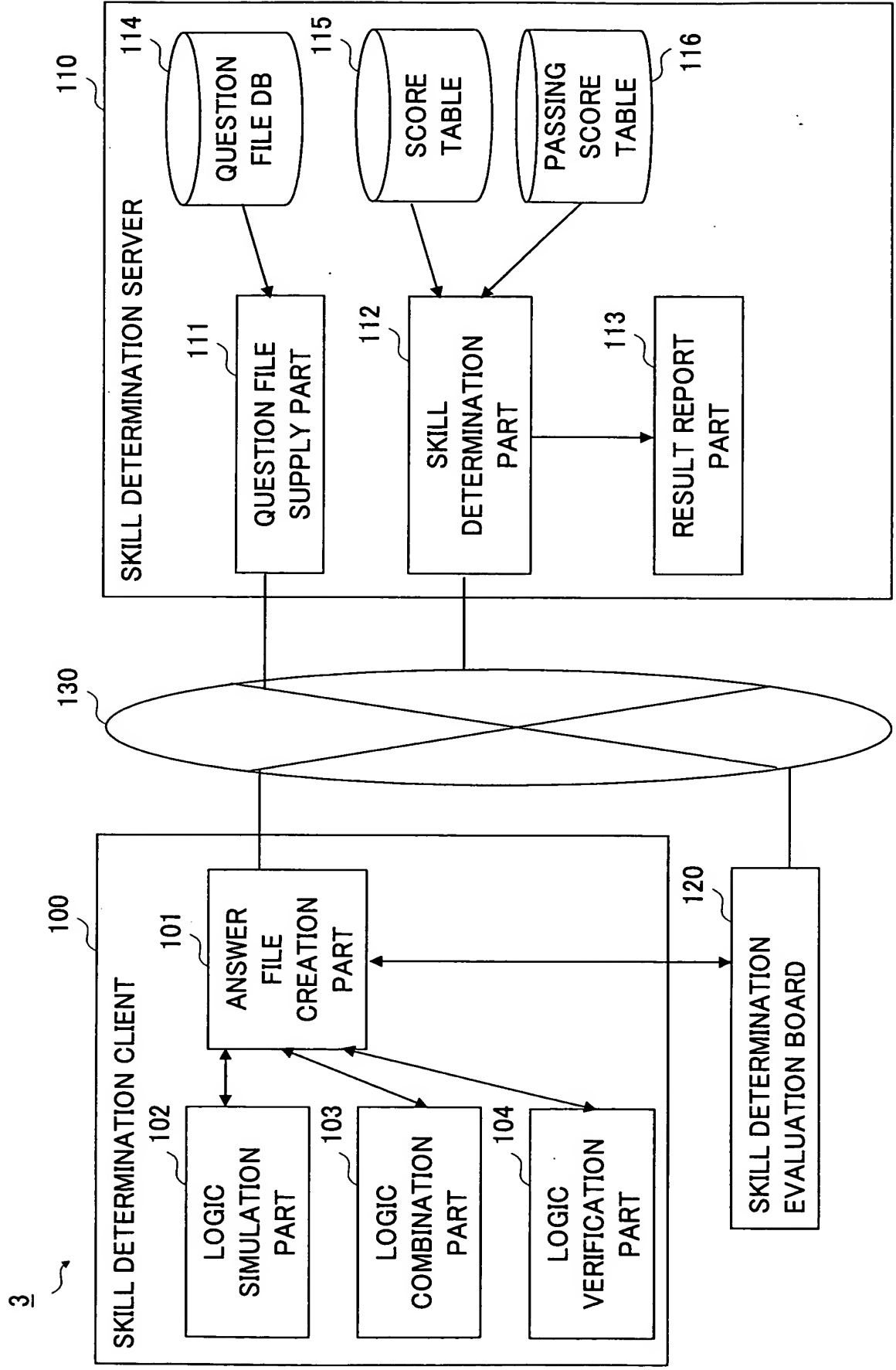


FIG.21

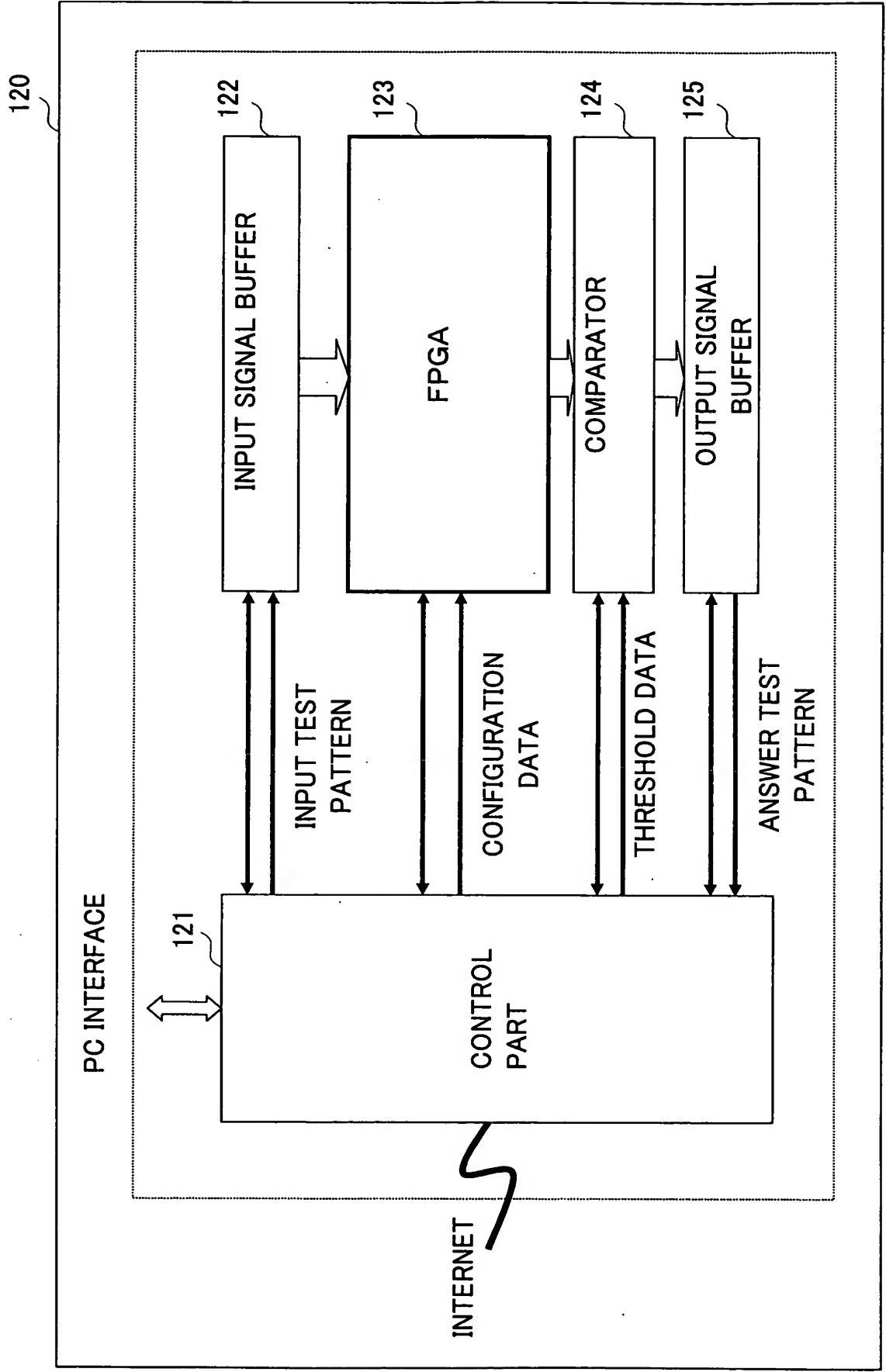


FIG.22

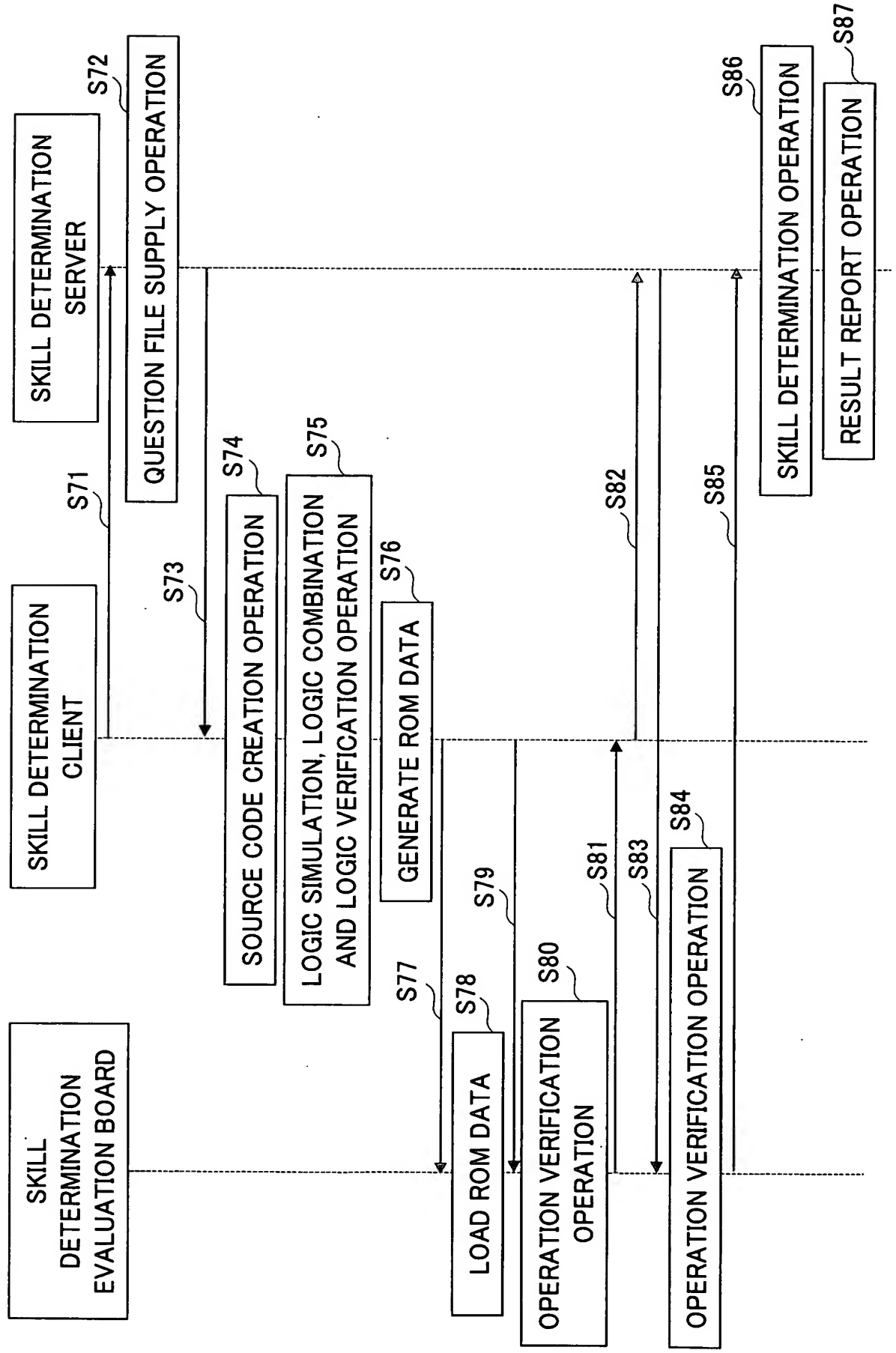


FIG.23

